

JEGYZÖKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Ebtenyésztés és tartás

Készítette: **Nyíri Beáta**

Neptunkód: **I40FDC**

Dátum:

A feladat leírása:

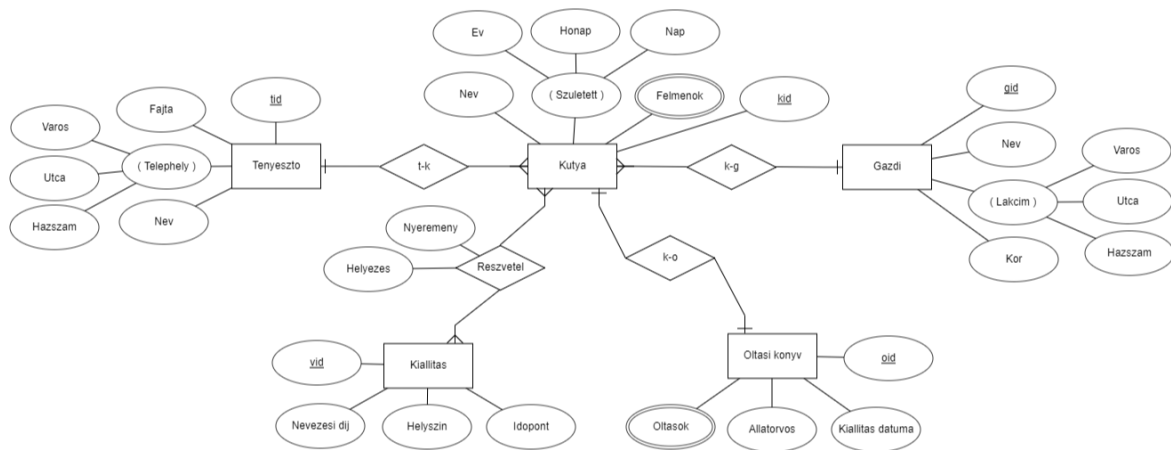
Az ebtenyésztés témaköre összetett. Több irányból is meg lehet közelíteni attól függően, hogy ki melyik részét találja fontosnak. A féléves feladathoz én úgy döntöttem, hogy a tenyésztők, kutyák, kutyatartók (gazdik), versenyek és oltási könyvek kapcsolatát mutatom be.

1. feladat

1a) Az adatbázis ER modell

A feladathoz egy olyan modellt készítettem, amely az ebtenyésztés és tartás témakörén belül mutat be egyedeket, tulajdonságokat és kapcsolatokat.

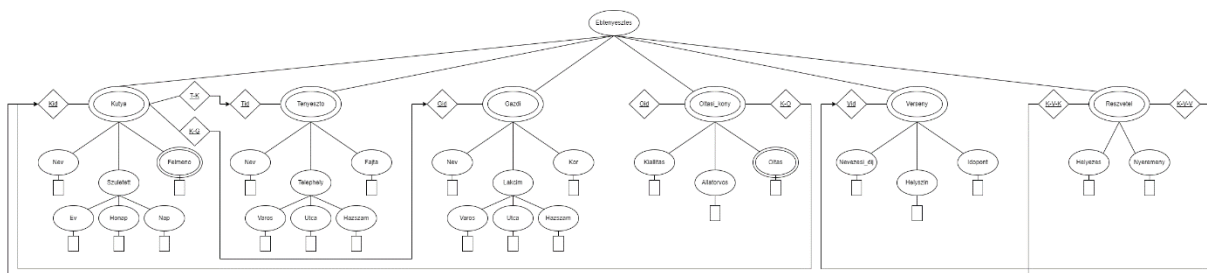
Ehhez 5 egyedet készítettem (Tenyésztő, Kutya, Gazdi, Verseny, Oltási könyv), mindegyik egyedhez kapcsoltam minimum 4 attribútumot (kulcs mind az öt esetében van, összetett és többértékű nem mindegyiknél). Az egyedek között létezik 1:1 (k-o), 1:N (t-k, k-g) és M:N kapcsolat (Részvétel) is, a M:N kapcsolat pedig rendelkezik két tulajdonsággal.



1b) Az adatbázis konvertálása XDM modellre

Az XDM modellhez készítettem egy gyökérelmet (Ebtenyesztes) és az ER modell egyedei lettek a gyermekelemei. Az elemek alá tettem az elemtulajdonságokat, amik alatt az értéket jelző téglalapok találhatóak.

A kulcsokat attribútummá konvertáltam (rombusz), és az elsődleges kulcsok mellett megjelenítettem az idegen kulcsokat is, amik a kapcsolatokból lettek. Végül összekötöttem a megfelelő idegen kulcsokat a megfelelő elsődleges kulcsokkal.



1c) Az XDM modell alapján XML dokumentum készítése

Az XML dokumentumban létrehoztam hat kutya példányt, amiknek mind három attribútuma lett, ezek a kid (a kutyák egyedi azonosítói), a t_k (idegen kulcs, ami összekapcsolja a megfelelő tenyésztővel) és a k_g (idegen kulcs a gazdihoz).

Tenyésztőből három példányt hoztam létre egy-egy attribútummal (egyedi azonosító), gazdiból és versenyből szintén hármat egy attribútummal (egyedi azonosító).

Oltási könyvből ugyanannyi példányt készítettem, mint kutyából és elláttam egy id attribútummal és egy idegen kulccsal (k_o) ami egy oltási könyvet összeköt egy kutyával.

A versenyek és kutyák összekapcsolásához pedig létrehoztam öt részvétel példányt, amiknek csak olyan attribútumai vannak, amik egy kutyára (k_v_k) vagy egy versenyre (k_v_v) hivatkoznak.

Ezek után feltöltöttem a tulajdonságokat értékekkel.

Kutyák:

```
<?xml version="1.0" encoding="UTF-8"?>
<ebtenyesztes>
  <!-- Kutyak -->
  <kutya kid="k1" t_k="t1" k_g="g1">
    <nev>Jumbo</nev>
    <szuletett>
      <ev>2022</ev>
      <honap>július</honap>
      <nap>2</nap>
    </szuletett>
    <felmeno>Vanilla</felmeno>
    <felmeno>Rinky-Dink Showman</felmeno>
    <felmeno>Ruby Red Viky Pei Deja Vu</felmeno>
    <felmeno>Captain Nemo Sleepland</felmeno>
  </kutya>

  <kutya kid="k2" t_k="t2" k_g="g2">
    <nev>Anasztasia</nev>
    <szuletett>
      <ev>2018</ev>
      <honap>július</honap>
      <nap>8</nap>
    </szuletett>
    <felmeno>Jazzy'pei Monty Python The Black Knight</felmeno>
    <felmeno>Gladys</felmeno>
    <felmeno>Hellboy Junior Gold Panonia des Chandagueres</felmeno>
    <felmeno>ESPARTINAS DE AIRES DE AL ANDALUS</felmeno>
  </kutya>

  <kutya kid="k3" t_k="t1" k_g="g2">
    <nev>Angel Pei Lucifer</nev>
    <szuletett>
      <ev>2020</ev>
      <honap>július</honap>
      <nap>4</nap>
    </szuletett>
    <felmeno>Ruby Red Viky Pei Moby Dick</felmeno>
    <felmeno>Cherry Pei Henna</felmeno>
    <felmeno>Shira Victory Dolce Vita</felmeno>
  </kutya>
```

```
<kutya kid="k4" t_k="t1" k_g="g3">
  <nev>Nutella</nev>
  <szuletett>
    <ev>2020</ev>
    <honap>szeptember</honap>
    <nap>20</nap>
  </szuletett>
  <felmeno>Leroy Hercules</felmeno>
  <felmeno>Ruby Red Viky Pei Celebrity</felmeno>
  <felmeno>Shira Victory Dolce Vita</felmeno>
</kutya>

<kutya kid="k5" t_k="t3" k_g="g3">
  <nev>Lancelot Zuzkin Sen</nev>
  <szuletett>
    <ev>2013</ev>
    <honap>december</honap>
    <nap>13</nap>
  </szuletett>
  <felmeno>Matgo Magic Aussie</felmeno>
  <felmeno>Olinka Magic Sonet</felmeno>
</kutya>

<kutya kid="k6" t_k="t3" k_g="g3">
  <nev>Portos Zuzkin Sen</nev>
  <szuletett>
    <ev>2018</ev>
    <honap>január</honap>
    <nap>20</nap>
  </szuletett>
  <felmeno>Sheik the Wild Waker de Tianxia</felmeno>
  <felmeno>Lolotka Zuzkin Sen</felmeno>
</kutya>
```

Tenyésztők:

```
<!-- Tenyésztők -->
<tenyeszto tid="t1">
  <nev>Kosaras Viktória</nev>
  <telephely>
    <varos>Kaposvár</varos>
    <utca>Szőlősy út</utca>
    <hazszam>3</hazszam>
  </telephely>
  <fajta>Shar Pei</fajta>
</tenyeszto>

<tenyeszto tid="t2">
  <nev>Baka László</nev>
  <telephely>
    <varos>Balmazújváros</varos>
    <utca>Wesselényi utca</utca>
    <hazszam>13</hazszam>
  </telephely>
  <fajta>Puli</fajta>
</tenyeszto>

<tenyeszto tid="t3">
  <nev>Zuzana Totová</nev>
  <telephely>
    <varos>Rožňava</varos>
    <utca>Lesná utca</utca>
    <hazszam>55</hazszam>
  </telephely>
  <fajta>Ausztrál Juhászkutya</fajta>
</tenyeszto>
```

Gazdik:

```
<!-- Gazdik -->
<gazdi gid="g1">
  <nev>Nyíri Beáta</nev>
  <lakcim>
    <varos>Sajószentpéter</varos>
    <utca>Irinyi János utca</utca>
    <hazszam>9</hazszam>
  </lakcim>
  <kor>22</kor>
</gazdi>

<gazdi gid="g2">
  <nev>Kiss Géza</nev>
  <lakcim>
    <varos>Miskolc</varos>
    <utca>Egressy út</utca>
    <hazszam>23</hazszam>
  </lakcim>
  <kor>45</kor>
</gazdi>

<gazdi gid="g3">
  <nev>Árpás Mónika</nev>
  <lakcim>
    <varos>Balatonfüred</varos>
    <utca>Mogyoró utca</utca>
    <hazszam>1</hazszam>
  </lakcim>
  <kor>32</kor>
</gazdi>
```

Oltási könyvek:

```
<!-- Oltási könyvek -->
<oltasi_konyv oid="o1" k_o="k1">
  <kiallitas>2022-08-15</kiallitas>
  <allatorvos>Dr. Jackó Géza</allatorvos>
  <oltas>Parvovírus</oltas>
  <oltas>Kombinált</oltas>
  <oltas>Veszétség1</oltas>
</oltasi_konyv>

<oltasi_konyv oid="o2" k_o="k2">
  <kiallitas>2018-08-10</kiallitas>
  <allatorvos>Dr. Kiss Krisztina</allatorvos>
  <oltas>Parvovírus</oltas>
  <oltas>Kombinált</oltas>
  <oltas>Veszétség1</oltas>
  <oltas>Veszétség2</oltas>
</oltasi_konyv>

<oltasi_konyv oid="o3" k_o="k3">
  <kiallitas>2020-08-30</kiallitas>
  <allatorvos>Dr. Major András</allatorvos>
  <oltas>Parvovírus</oltas>
  <oltas>Kombinált</oltas>
  <oltas>Veszétség1</oltas>
  <oltas>Veszétség2</oltas>
</oltasi_konyv>
```

```
<oltasi_konyv oid="o4" k_o="k4">
  <kiallitas>2020-10-11</kiallitas>
  <allatorvos>Dr. Fülöp Andrea</allatorvos>
  <oltas>Parvovírus</oltas>
  <oltas>Kombinált</oltas>
  <oltas>Veszétség1</oltas>
  <oltas>Veszétség2</oltas>
</oltasi_konyv>

<oltasi_konyv oid="o5" k_o="k5">
  <kiallitas>2014-01-23</kiallitas>
  <allatorvos>Dr. Vásárhelyi Géza</allatorvos>
  <oltas>Parvovírus</oltas>
  <oltas>Kombinált</oltas>
  <oltas>Veszétség1</oltas>
  <oltas>Veszétség2</oltas>
  <oltas>Veszétség3</oltas>
</oltasi_konyv>

<oltasi_konyv oid="o6" k_o="k6">
  <kiallitas>2018-02-28</kiallitas>
  <allatorvos>Dr. Nagy Zoltán</allatorvos>
  <oltas>Parvovírus</oltas>
  <oltas>Kombinált</oltas>
  <oltas>Veszétség1</oltas>
  <oltas>Veszétség2</oltas>
</oltasi_konyv>
```

Versenyek:

```
<!-- Versenyek -->
<verseny vid="v1">
  <nevezesi_dij>10.000 Ft</nevezesi_dij>
  <helyszin>Budapest</helyszin>
  <idopont>2022-08-08</idopont>
</verseny>

<verseny vid="v2">
  <nevezesi_dij>7.000 Ft</nevezesi_dij>
  <helyszin>Debrecen</helyszin>
  <idopont>2019-11-01</idopont>
</verseny>

<verseny vid="v3">
  <nevezesi_dij>20 €</nevezesi_dij>
  <helyszin>Kassa</helyszin>
  <idopont>2015-03-20</idopont>
</verseny>
```

Résztvételek:

```
<!-- Versenven valo reszvetelek -->
<reszvetel k_v_k="k2" k_v_v="v1">
  <helyezes>1</helyezes>
  <nyeremeny>70.000 Ft</nyeremeny>
</reszvetel>

<reszvetel k_v_k="k3" k_v_v="v1">
  <helyezes>3</helyezes>
  <nyeremeny>30.000 Ft</nyeremeny>
</reszvetel>

<reszvetel k_v_k="k5" k_v_v="v2">
  <helyezes>2</helyezes>
  <nyeremeny>50.000 Ft</nyeremeny>
</reszvetel>

<reszvetel k_v_k="k6" k_v_v="v2">
  <helyezes>3</helyezes>
  <nyeremeny>25.000 Ft</nyeremeny>
</reszvetel>

<reszvetel k_v_k="k5" k_v_v="v3">
  <helyezes>2</helyezes>
  <nyeremeny>130 €</nyeremeny>
</reszvetel>

</ebtenyesztes>
```

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok, ref, key, keyref, speciális elemek)

Az XSD dokumentumban először definiáltam a saját típusaimat, majd ezekkel hoztam létre az elementeket. Ezután létrehoztam az elsődleges kulcsokat, és végül az idegen kulcsokat.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!-- Saját típusok -->
  <xs:simpleType name="kutyaTípus">
    <xs:restriction base="xs:string">
      <xs:minLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tenyesztoTípus">
    <xs:restriction base="xs:string">
      <xs:minLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="gazdiTípus">
    <xs:restriction base="xs:string">
      <xs:minLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
```

```

<xs:simpleType name="oltasikonyvTipus">
  <xs:restriction base="xs:string">
    <xs:minLength value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="versenyTipus">
  <xs:restriction base="xs:string">
    <xs:minLength value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="reszvetelTipus">
  <xs:restriction base="xs:string">
    <xs:minLength value="2"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:element name="ebtenyesztes">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="kutya" type="kutyaTipus" maxOccurs="unbounded"/>
      <xs:element name="tenyeszto" type="tenyesztoTipus" maxOccurs="unbounded"/>
      <xs:element name="gazdi" type="gazdiTipus" maxOccurs="unbounded"/>
      <xs:element name="oltasikonyv" type="oltasikonyvTipus" maxOccurs="unbounded"/>
      <xs:element name="verseny" type="versenyTipus" maxOccurs="unbounded"/>
      <xs:element name="reszvetel" type="reszvetelTipus" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!-- Elsődleges Kulcsok -->
<xs:key name="kutya_kulcs">
  <xs:selector xpath="kutya" />
  <xs:field xpath="@kid" />
</xs:key>
<xs:key name="tenyeszto_kulcs">
  <xs:selector xpath="tenyeszto" />
  <xs:field xpath="@tid" />
</xs:key>
<xs:key name="gazdi_kulcs">
  <xs:selector xpath="gazdi" />
  <xs:field xpath="@gid" />
</xs:key>
<xs:key name="oltasikonyv_kulcs">
  <xs:selector xpath="oltasikonyv" />
  <xs:field xpath="@oid" />
</xs:key>
<xs:key name="verseny_kulcs">
  <xs:selector xpath="verseny" />
  <xs:field xpath="@vid" />
</xs:key>

```

```

<!-- Idegen Kulcsok-->
<xs:keyref refer="kutya_kulcs" name="kutya_idegen_kulcs">
  <xs:selector xpath="reszvetel" />
  <xs:field xpath="@k_v_k" />
</xs:keyref>
<xs:keyref refer="verseny_kulcs" name="verseny_idegen_kulcs">
  <xs:selector xpath="reszvetel" />
  <xs:field xpath="@k_v_v" />
</xs:keyref>
<xs:keyref refer="tenyeszto_kulcs" name="kutya_tenyeszto_idegen_kulcs">
  <xs:selector xpath="tenyeszto" />
  <xs:field xpath="@k_t" />
</xs:keyref>
<xs:keyref refer="gazdi_kulcs" name="kutya_gazdi_idegen_kulcs">
  <xs:selector xpath="gazdi" />
  <xs:field xpath="@k_g" />
</xs:keyref>
<xs:keyref refer="kutya_kulcs" name="oltasikonyv_idegen_kulcs">
  <xs:selector xpath="kutya" />
  <xs:field xpath="@k_o" />
</xs:keyref>
</xs:schema>

```

2. feladat

2a) adatolvasás

A DOMRead fájlban egyesével kiíratam minden elemnek az értékeit a konzolra, és a konzolon megjelent adatokat egy output.txt fájlba mentettem el a következő két sor segítségével:

```

PrintStream myconsole = new PrintStream(new File("eleresi_utvonal\\output.txt"));
System.setOut(myconsole);

```

```

1 package hu.dompars.e.i40fdc;
2
3 import org.w3c.dom.Document;
4
5
6 public class DOMReadI40FDC {
7
8
9     public static void main(String[] args) {
10
11         DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
12
13         try {
14
15             // TXT fájlba mentes
16             //PrintStream myconsole = new PrintStream(new File("C:\\Users\\Admin\\OneDrive\\
17             //System.setOut(myconsole);
18
19             // parse XML file
20             DocumentBuilder db = dbf.newDocumentBuilder();
21
22             Document doc = db.parse(new File("XMLi40fdc.xml"));
23
24             System.out.println("Root Element : " + doc.getDocumentElement().getNodeName());
25             System.out.println("-----");
26

```


Kutyák:

```
// ----- Kutyák -----
NodeList listKutya = doc.getElementsByTagName("kutya");

for (int i = 0; i < listKutya.getLength(); i++) {
    Node node = listKutya.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        // attributumok
        String kid = element.getAttribute("kid");
        String k_t = element.getAttribute("k_t");
        String k_g = element.getAttribute("k_g");
        // szoveg kiolvasasa
        String nev = element.getElementsByTagName("nev").item(0).getTextContent();
        String ev = element.getElementsByTagName("ev").item(0).getTextContent();
        String honap = element.getElementsByTagName("honap").item(0).getTextContent();
        String nap = element.getElementsByTagName("nap").item(0).getTextContent();
        // tobberteku elemekhez lista
        NodeList felmenok = element.getElementsByTagName("felmeno");
        // adatok kiiratasa
        System.out.println("Current Element : " + node.getNodeName());
        System.out.println("ID : " + kid);
        System.out.println("Tenyeszto ID : " + k_t);
        System.out.println("Gazdi ID : " + k_g);
        System.out.println("Nev : " + nev);
        System.out.println("Szuletett : " + ev + " " + honap + " " + nap + ".");
        for (int j=0; j<felmenok.getLength(); j++) {
            System.out.println("Felmeno : " + felmenok.item(j).getTextContent());
        }
        System.out.println();
    }
}
System.out.println("-----\n");
```

Tenyésztők:

```
// ----- Tenyésztők -----
NodeList listTenyeszto = doc.getElementsByTagName("tenyeszto");

for (int i = 0; i < listTenyeszto.getLength(); i++) {
    Node node = listTenyeszto.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        // attributumok
        String tid = element.getAttribute("tid");
        // szoveg kiolvasasa
        String nev = element.getElementsByTagName("nev").item(0).getTextContent();
        String varos = element.getElementsByTagName("varos").item(0).getTextContent();
        String utca = element.getElementsByTagName("utca").item(0).getTextContent();
        String hazszam = element.getElementsByTagName("hazszam").item(0).getTextContent();
        String fajta = element.getElementsByTagName("fajta").item(0).getTextContent();
        // adatok kiiratasa
        System.out.println("Current Element : " + node.getNodeName());
        System.out.println("ID : " + tid);
        System.out.println("Nev : " + nev);
        System.out.println("Telephely : " + varos + ", " + utca + " " + hazszam + ".");
        System.out.println("Tenyesztett fajta : " + fajta);
        System.out.println();
    }
}
System.out.println("-----\n");
```


Gazdik:

```
// ----- Gazdik -----
NodeList listGazdi = doc.getElementsByTagName("gazdi");

for (int i = 0; i < listGazdi.getLength(); i++) {
    Node node = listGazdi.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        // attributumok
        String gid = element.getAttribute("gid");
        // szoveg kiolvasasa
        String nev = element.getElementsByTagName("nev").item(0).getTextContent();
        String varos = element.getElementsByTagName("varos").item(0).getTextContent();
        String utca = element.getElementsByTagName("utca").item(0).getTextContent();
        String hazszam = element.getElementsByTagName("hazszam").item(0).getTextContent();
        String kor = element.getElementsByTagName("kor").item(0).getTextContent();
        //adatok kiiratasa
        System.out.println("Current Element : " + node.getNodeName());
        System.out.println("ID : " + gid);
        System.out.println("Nev : " + nev);
        System.out.println("Lakcim : " + varos + ", " + utca + " " + hazszam + ".");
        System.out.println("Kor : " + kor);
        System.out.println();
    }
}
System.out.println("-----\n");
```

Oltasi könyvek:

```
// ----- Oltasi könyvek -----
NodeList listOltasi = doc.getElementsByTagName("oltasi_konyv");

for (int i = 0; i < listOltasi.getLength(); i++) {
    Node node = listOltasi.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        // attributumok
        String oid = element.getAttribute("oid");
        String k_o = element.getAttribute("k_o");
        // szoveg kiolvasasa
        String kiallitas = element.getElementsByTagName("kiallitas").item(0).getTextContent();
        String allatorvos = element.getElementsByTagName("allatorvos").item(0).getTextContent();
        //tobbenteku elemekhez lista
        NodeList oltasok = element.getElementsByTagName("oltas");
        //adatok kiiratasa
        System.out.println("Current Element : " + node.getNodeName());
        System.out.println("ID : " + oid);
        System.out.println("Kutya ID : " + k_o);
        System.out.println("Kiallitas : " + kiallitas);
        for (int j=0; j<oltasok.getLength(); j++) {
            System.out.println("Oltas : " + oltasok.item(j).getTextContent());
        }
        System.out.println();
    }
}
System.out.println("-----\n");
```

Versenyek:

```
// ----- Versenyek -----
NodeList listVerseny = doc.getElementsByTagName("verseny");

for (int i = 0; i < listVerseny.getLength(); i++) {
    Node node = listVerseny.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        // attributumok
        String vid = element.getAttribute("vid");
        // szoveg kiolvasasa
        String dij = element.getElementsByTagName("nevezesi_dij").item(0).getTextContent();
        String helyszin = element.getElementsByTagName("helyszin").item(0).getTextContent();
        String idopont = element.getElementsByTagName("idopont").item(0).getTextContent();
        // adatok kiiratasa
        System.out.println("Current Element : " + node.getNodeName());
        System.out.println("ID : " + vid);
        System.out.println("Nevezesi dij : " + dij);
        System.out.println("Helyszin : " + helyszin);
        System.out.println("Idopont : " + idopont);
        System.out.println();
    }
}
System.out.println("-----\n");
```

Résztveletek:

```
// ----- Versenyen valo reszvetelek -----
NodeList listReszvetel = doc.getElementsByTagName("reszvetel");

for (int i = 0; i < listReszvetel.getLength(); i++) {
    Node node = listReszvetel.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        // attributumok
        String k_v_k = element.getAttribute("k_v_k");
        String k_v_v = element.getAttribute("k_v_v");
        // szoveg kiolvasasa
        String helyezes = element.getElementsByTagName("helyezes").item(0).getTextContent();
        String nyeremeny = element.getElementsByTagName("nyeremeny").item(0).getTextContent();
        // adatok kiiratasa
        System.out.println("Current Element : " + node.getNodeName());
        System.out.println("Kutya ID : " + k_v_k);
        System.out.println("Verseny ID : " + k_v_v);
        System.out.println("Helyezes : " + helyezes);
        System.out.println("Nyeremeny : " + nyeremeny);
        System.out.println();
    }
}

} catch (ParserConfigurationException | SAXException | IOException e) {
    e.printStackTrace();
}
}
```


A konzolom a következő jelenik meg:

```
Root Element : ebtenyesztes
-----
Current Element : kutya
ID : k1
Tenyeszto ID : t1
Gazdi ID : g1
Nev : Jumbo
Szuletett : 2022. július 2.
Felmeno : Vanilla
Felmeno : Rinky-Dink Showman
Felmeno : Ruby Red Viky Pei Deja Vu
Felmeno : Captain Nemo Sleepland

Current Element : kutya
ID : k2
Tenyeszto ID : t2
Gazdi ID : g2
Nev : Anastasia
Szuletett : 2018. július 8.
Felmeno : Jazzy'pei Monty Python The Black Knight
Felmeno : Gladys
Felmeno : Hellboy Junior Gold Panonia des Chandagueres
Felmeno : ESPARTINAS DE AIRES DE AL ANDALUS

Current Element : kutya
ID : k3
Tenyeszto ID : t1
Gazdi ID : g2
```

Az output.txt fájlban ugyanez van elmentve:

 output.txt – Jegyzettömb

Fájl Szerkesztés Formátum Nézet Súgó

```
Root Element : ebtenyesztes
-----
Current Element : kutya
ID : k1
Tenyeszto ID : t1
Gazdi ID : g1
Nev : Jumbo
Szuletett : 2022. július 2.
Felmeno : Vanilla
Felmeno : Rinky-Dink Showman
Felmeno : Ruby Red Viky Pei Deja Vu
Felmeno : Captain Nemo Sleepland

Current Element : kutya
ID : k2
Tenyeszto ID : t2
Gazdi ID : g2
Nev : Anastasia
Szuletett : 2018. július 8.
Felmeno : Jazzy'pei Monty Python The Black Knight
Felmeno : Gladys
Felmeno : Hellboy Junior Gold Panonia des Chandagueres
Felmeno : ESPARTINAS DE AIRES DE AL ANDALUS

Current Element : kutya
ID : k3
```

2b) adatmódosítás

Ehhez a feladathoz az XML fájlban 4 módosítást végeztem. Ezek a módosítások a következők:

- Egy kutya ID-jének a módosítása,
- Egy kutya nevének a módosítása,
- Egy kutya egyik felmenőjének a módosítása,
- Oltási könyvek törlése.

Ezeket a módosításokat nem mentettem el, csak kiirattam a konzolra.

```
package hu.domparse.i40fdc;

import java.io.File;

public class DOMModifyI40FDC {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {

        try {
            //file nev megadása, parszolás
            File inputFile = new File("XMLi40fdc.xml");
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);

            //a második és a harmadik kutya adatainak mentése
            Node kutya2 = doc.getElementsByTagName("kutya").item(1);
            Node kutya3 = doc.getElementsByTagName("kutya").item(2);
            //gyökerelem mentése
            Node ebtenyesztes = doc.getFirstChild();
```

```
            //kutya ID modositasa
            NamedNodeMap attr = kutya2.getAttributes();
            Node nodeAttr = attr.getNamedItem("kid");
            nodeAttr.setTextContent("k7");

            //a második kutya nevének modositasa
            NodeList list = kutya2.getChildNodes();
            for (int i = 0; i < list.getLength(); i++) {
                Node node = list.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element elem = (Element) node;
                    if ("nev".equals(elem.getNodeName())) {
                        if("Anasztasia".equals(elem.getTextContent())) {
                            elem.setTextContent("Snowglobe");
                        }
                    }
                }
            }
        }
    }
}
```

```

//a harmadik kutya egyik felmenoiének modositasa
NodeList list1 = kutya3.getChildNodes();
for (int i = 0; i < list1.getLength(); i++) {
    Node node = list1.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;
        if ("felmeno".equals(elem.getNodeName())) {
            if("Shira Victory Dolce Vita".equals(elem.getTextContent())) {
                elem.setTextContent("Shira Victory Sunshine");
            }
        }
    }
}

//oltasi konyvek torlese
NodeList childNodes = ebtenyesztes.getChildNodes();
for(int i = 0; i < childNodes.getLength(); i++) {
    Node node = childNodes.item(i);

    if("oltasi_konyv".equals(node.getNodeName()))
        ebtenyesztes.removeChild(node);
}

```

```

//megjelenites a consolon
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource(doc);
System.out.println("-----New File-----");
StreamResult consoleResult = new StreamResult(System.out);
transformer.transform(source, consoleResult);
}catch(Exception e) {
    e.printStackTrace();
}
}
}

```

2c) adatlekérdezés

Az utolsó feladathoz három lekérdezést készítettem, amelyek megadják...

- a versenyeken második helyezést elérők nyereményeit;
- a shar pei tenyésztőket;
- a 30 évnél idősebb gazdikat.

Az lekérdezések eredményeit végül a konzolra irattam ki.

```
package hu.domparsa.i40fdc;

import java.io.File;

public class DOMQueryI40FDC {

    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLi40fdc.xml");
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document doc = dBuilder.parse(xmlFile);
            doc.getDocumentElement().normalize();

            System.out.println("Második helyezettek nyereményei: \n");
            Competition(doc);
            System.out.println("-----");

            System.out.println("\nShar pei tenyésztők: \n");
            SharPeiBreeder(doc);
            System.out.println("-----");

            System.out.println("\n30 év feletti gazdik: \n");
            OlderThanThirty(doc);

        } catch (ParserConfigurationException | IOException | SAXException ex){
            System.out.println("Some error happened:\n"+ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```

```
private static String NormalizeText(String text) {
    text=text.replaceAll("\\n", " ");
    text=text.replaceAll("\\s+", " ");
    return text;
}

//második helyen végzetek nyeremenvei
private static void Competition(Document doc) {
    NodeList tenyesztok = doc.getElementsByTagName("reszvetel");
    for(int i = 0; i < tenyesztok.getLength(); i++) {
        Element elem =(Element)tenyesztok.item(i);
        NodeList childNodes = elem.getChildNodes();
        for(int j =0;j<childNodes.getLength();j++) {
            Node childNode = childNodes.item(j);
            if(childNode.getNodeName().equals("helyezes")) {
                if(Integer.parseInt(childNode.getTextContent())==2) {
                    PrintElement(elem);
                }
            }
        }
    }
}

}
```

```

//shar pei tenyeszto
private static void SharPeiBreeder(Document doc) {
    NodeList tenyesztok = doc.getElementsByTagName("tenyeszto");
    for(int i = 0; i < tenyesztok.getLength(); i++) {
        Element elem =(Element)tenyesztok.item(i);
        NodeList childNodes = elem.getChildNodes();
        for(int j =0;j<childNodes.getLength();j++) {
            Node childNode = childNodes.item(j);
            if(childNode.getNodeName().equals("fajta")) {
                if(childNode.getTextContent().equals("Shar Pei")) {
                    PrintElement(elem);
                }
            }
        }
    }
}

//30 ev feletti gazdik
private static void OlderThanThirty(Document doc) {
    NodeList gazdik = doc.getElementsByTagName("gazdi");
    for(int i = 0; i < gazdik.getLength(); i++) {
        Element elem =(Element)gazdik.item(i);
        NodeList childNodes = elem.getChildNodes();
        for(int j =0;j<childNodes.getLength();j++) {
            Node childNode = childNodes.item(j);
            if(childNode.getNodeName().equals("kor")) {
                if(Integer.parseInt(childNode.getTextContent())>=30) {
                    PrintElement(elem);
                }
            }
        }
    }
}

```

```

//adatok kiirasa konzolra
private static void PrintElement(Element elem) {
    String kid = elem.getAttribute("kid");
    String tid = elem.getAttribute("tid");
    String gid = elem.getAttribute("gid");

    if(kid != "") {
        System.out.println("ID: "+ kid);
    }else if(tid != "") {
        System.out.println("ID: "+ tid);
    } else if(gid != "") {
        System.out.println("ID: "+ gid);
    }

    String nodeContent="";
    NodeList childNodes = elem.getChildNodes();
    for(int j =0;j<childNodes.getLength() ; j++) {
        if(childNodes.item(j).getTextContent().trim()!="") {
            nodeContent = NormalizeText(childNodes.item(j).getTextContent().trim());
            System.out.println(childNodes.item(j).getNodeName()+": "+nodeContent);
        }
    }
}
}

```


Az eredmény:

Második helyezettek nyereményei:

helyezés: 2

nyeremeny: 50.000 Ft

helyezés: 2

nyeremeny: 130 €

Shar pei tenyésztők:

ID: t1

nev: Kosaras Viktória

telephely: Kaposvár, Szőlősy út, 3

fajta: Shar Pei

30 év feletti gazdik:

ID: g2

nev: Kiss Géza

lakcím: Miskolc, Egressy út, 23

kor: 45

ID: g3

nev: Árpás Mónika

lakcím: Balatonfüred, Mogyoró utca, 1

kor: 32